

Docket No.: B2745.0023/P0023
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Jonathan Dorfman

Application No.: 09/372,416

Group Art Unit: 2178

Filed: August 11, 1999

Examiner: T. V. Huynh

For: EASILY MODIFIABLE MACRO TAG FOR
INTERNET ADVERTISING

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF CHARLES R. MARTIN UNDER 37 CFR § 1.132

INTRODUCTION

(1) In this declaration I summarize my findings in response to questions posed to me on technical issues presented by U.S. Patent No. **6,516,338**, entitled "Apparatus and Accompanying Methods for implementing Network Servers for Use in Providing Interstitial Web Advertisements to a Client Computer", issued to **Landsman et al.**, on Feb. 4, 2003 (hereinafter referred to as the "Landsman Patent", "Landsman" or "Patent").

(2) In preparation of this declaration, I have read the entire Landsman Patent and examined the questions at issue that have been presented to me. It has been explained to me that a USPTO Examiner has cited Landsman as prior art to pending U.S. Patent Application Serial No. 09/372,416 (hereinafter "'416 Application").

Declaration of Charles R. Martin

1

However, in preparation of this declaration, I have not read the '416 Application, nor have I read the Office Action issued by the Examiner. As such, this declaration only addresses the issues raised with respect to the Landsman Patent, but does not directly discuss the Examiner's assertions nor does it discuss the patentability of the '416 Application.

(3) In this declaration I question some of the terminology used in Landsman Patent, and I also assert that some statements in the Landsman Patent are necessarily misleading and incorrect. Such findings were derived by reading the Patent as a whole and making deductions based on the whole subject matter and in view of the general underlying technologies as known to one skilled in the art.

QUALIFICATIONS

(4) I have been a professional programmer, software engineer, and computer scientist since 1969. I hold a Master of Science in Computer Science (with minors in Electrical Engineering and Mathematics) from Duke University (1988), and have completed all PhD requirements in Computer Science except for dissertation both at Duke University in 1990 and at the University of North Carolina at Chapel Hill in 1991. I was a Senior Consultant with IBM in the Object Practice, and one of the first people to implement web-enabled e-commerce systems for customers such as Cummins Engine Company, and SABRE Inc. I then was a Senior Java Architect for Sun Microsystems from 1998 to 2001, and consulted worldwide in Java systems, Java architecture, and e-commerce systems, including web-based systems. As a consultant, my clients have included the Department of Defense (via DARPA and the Department of the Navy), NASA, McGraw-Hill, the Canada Trust Bank, and about 50 other firms.

However, in preparation of this declaration, I have not read the '416 Application, nor have I read the Office Action issued by the Examiner. As such, this declaration only addresses the issues raised with respect to the Landsman Patent, but does not directly discuss the Examiner's assertions nor does it discuss the patentability of the '416 Application.

(3) In this declaration I question some of the terminology used in Landsman Patent, and I also assert that some statements in the Landsman Patent are necessarily misleading and incorrect. Such findings were derived by reading the Patent as a whole and making deductions based on the whole subject matter and in view of the general underlying technologies as known to one skilled in the art.

QUALIFICATIONS

(4) I have been a professional programmer, software engineer, and computer scientist since 1969. I hold a Master of Science in Computer Science (with minors in Electrical Engineering and Mathematics) from Duke University (1988), and have completed all PhD requirements in Computer Science except for dissertation both at Duke University in 1990 and at the University of North Carolina at Chapel Hill in 1991. I was a Senior Consultant with IBM in the Object Practice, and one of the first people to implement web-enabled e-commerce systems for customers such as Cummins Engine Company, and SABRE Inc. I then was a Senior Java Architect for Sun Microsystems from 1998 to 2001, and consulted worldwide in Java systems, Java architecture, and e-commerce systems, including web-based systems. As a consultant, my clients have included the Department of Defense (via DARPA and the Department of the Navy), NASA, McGraw-Hill, the Canada Trust Bank, and about 50 other firms.

SUMMARY OF THE LANDSMAN PATENT

(5) Landsman teaches a system with techniques for dynamically displaying advertising content in a network based client/server environment. With Landsman's system, advertisements are downloaded from an advertising server to a browser, and are displayed on an interstitial basis, in response to the click stream generated by the user. Each web page utilizing the advertising system includes an advertising tag embedded within the HTML code of the web page. The advertising tag instantiates the automatic downloading and display of an advertisement.

(6) The advertising tag contains two components. The first component is a reference to a JavaScript file, which when downloaded and executed by a browser, dynamically inserts into the HTML code of the web page a combination of applet tags in lieu of the advertising tag. This JavaScript file resides on an 'agent server' along with components of the advertising agent that are downloaded to the client PC. When the browser executes the applet tags it causes the downloading from the agent server of an applet named the Transition Sensor applet (also referred to as the "TSA") and the browser then launches the TSA. Once the TSA is downloaded into the browser cache, it remains persistent throughout different web pages and websites and upon subsequent browser sessions, and is available for automatic loading from the browser cache. The applet tags will only cause the downloading of the TSA when it is not resident in the browser cache or when an updated version is required.

(7) The second component of the ad tag is the URL of a particular Ad Management System (also referred to as the "AMS"). The AMS is a third party advertising server containing specific media for advertisements and accompanying player files.

(8) When a user requests a web page that includes an embedded advertising tag,

Declaration of Charles R. Martin

3

the browser executes the JavaScript file referenced in the ad tag, which invokes the dynamic writing of the applet tags. The applet tags are then parsed by the browser, and further invoke the loading of the TSA. When the TSA is executed it causes a second applet named the AdController applet to be downloaded to the client computer from the agent server to the extent necessary, and the client computer then loads the AdController applet. The TSA creates an applet registry for inter-applet communication, and writes entries for both the TSA and the AdController applet.

(9) The TSA then passes the URL of the AMS as referenced in the advertising tag to the Ad Controller applet. The AdController applet in turn sends a request to the AMS for an advertisement. The AMS selects an advertisement and causes the AdController applet to begin downloading the advertisement. More specifically, for each advertisement on the AMS there is a unique file named an AdDescriptor file. The AdDescriptor file contains a list of media files associated with each advertisement. Once the AdController receives the AdDescriptor file, it starts the process of downloading all the files as listed in the AdDescriptor file. When the AdController receives the complete set of media files associated with an advertisement, the advertisement becomes available for display by the client PC.

(10) The TSA monitors the click stream of the user. When the TSA detects the browser's transition to a new web page it signals the AdController applet to play an advertisement previously cached by the AdController applet. The advertisements are played sequentially in the order of a play queue.

QUESTIONS AT ISSUE

- (11) The questions at issue are as follows:
- a. Is a reference to the AMS included in the Applet Tags?
 - b. Is a reference to the AMS included in the JavaScript file?
 - c. Is a reference to the AMS included in the Transition Sensor Applet?

Declaration of Charles R. Martin

4

DISCUSSION

The Applet Tags and the Transition Sensor Applet are separate elements.

(12) As an initial matter I note the following misuse of certain terminology by Landsman. A Java applet is a term of art, which denotes a compiled executable file that may be downloaded and launched by a web browser. A Java applet can request information from a remote server computer, but for security purposes the Java applet is limited to retrieving information from the same domain server as the Java applet resides on. A JavaScript file includes JavaScript script, and is not a compiled executable, rather the script is parsed by a java-enabled browser line by line, similarly to the parsing of a conventional HTML coded web page. A web page may also include an embedded JavaScript, also known as an ECMAScript, which is a technique for automatically rewriting the HTML of a web page. With this technique, the HTML of the web page includes a reference to a JavaScript file. The browser parses the HTML code, which invokes the parsing of the JavaScript file. The JavaScript file then causes the HTML code of the web page to be rewritten. Thereafter, the browser parses the rewritten HTML code and outputs the corresponding web page. This technique is also referred to as "dynamic HTML" and is utilized by Landsman (Col. 17 line 37 to Col 18 line 35), as I hereby emphasize.

(13) As noted above, Landsman describes an advertising tag that includes a reference to a JavaScript file. The advertising tag is in essence a fragment of HTML code that is exemplified in Table 1 in the Landsman Patent. When the JavaScript file is loaded and executed by the browser it dynamically overwrites the HTML code of the advertising tag with a different fragment of HTML code. This rewritten fragment of HTML code includes a reference to a java applet, i.e. the TSA. Landsman refers to the rewritten fragment of HTML as 'applet tags', because it comprises a reference to the TSA and a set of values, which are passed to the TSA. The applet tags are shown in

Declaration of Charles R. Martin

5

Col. 1 in Table 2. The value CODE defines the Java executable for the Transition Sensor applet. The following table illustrates the rewriting effect in Landsman:

Before Rewriting	After Rewriting
<pre> <p> Lorem ipsum dolor sit amet, condimentum ut, porta non. Taciti vestibulum imperdiet, felis nulla, at tristique et. </p> <SCRIPT> SRC=http://unicast.com/loadad.js> AdServer="http://AdManagement System" </SCRIPT> <p> Pretium sit. Et sed dictum. Vel tellus, ultrices sed, aenean velit quis. Feugiat ultrices, sed libero, in lobortis a. Commodo vitae commodo, amet purus, orci neque nec. </p> </pre>	<pre> <p> Lorem ipsum dolor sit amet, condimentum ut, porta non. Taciti vestibulum imperdiet, felis nulla, at tristique et. </p> <applet> code="com.unicast.adcontroller.tools.TransitionSensor" codebase="http://www.unicast.com/java/clases/" align="baseline" width="0" height="0" name="TransitionSensor" archive=adcontroller.jar"> <param name="adURL" value="http://www.unicast.com/media/fireworks_01_ad_descriptor.txt"> </applet> <p> Pretium sit. Et sed dictum. Vel tellus, ultrices sed, aenean velit quis. Feugiat ultrices, sed libero, in lobortis a. Commodo vitae commodo, amet purus, orci neque nec. </p> </pre>

(14) Thus, as has been explained, the 'applet tags', refer to the rewritten fragment of HTML code in the web page, and the Transition Sensor applet is a distinct executable file, referred to by the applet tags. As such, we must note that Landsman in numerous instances erroneously refers to the applet tags as being the same as the Transition Sensor Applet. For example, in Figure 2B, the applet tags (numeral 210) are captioned "Transition Sensor Applet", and the specification likewise refers to numeral 210 as the TSA or as an applet. Consider also the following passages in Landsman (emphasis added):

One portion of the advertising tag... downloads a JavaScript file ... This file... [is] executed, as a script...[und] substitute[s] applet tags, dynamically written by the script, into the referring web page in lieu of advertising tag 40 so as to form a modified web page.... Collectively, these applet tags form Transition Sensor applet 210. (Col 17 line 57 to Col 18 line 6).

The value of attribute CODE in the Transition Sensor applet specifies a Java executable that will be executed by the client browser, when it renders this applet, to launch the Transition Sensor. (Col 18 lines 24 to 27).

Declaration of Charles R. Martin

6

If advertisements are to play on client browsers... that do not support dynamic writing of applet tags, then applet 210 would need to be inserted by content providers into each referring web page in lieu of advertising tag 40. Unfortunately, Transition Sensor applet 210 identifies both the agent server...Hence, we see little, and very shortly essentially no need, to embed applet 210 into any referring web pages...(Col 18 lines 48 to 65)

(15) Hence, it is important to point out that Landsman incorrectly refers to the 'applet tags' as an applet. Rather the applet tags are part of the rewritten HTML code, which invoke the Java applet through the value 'CODE'.

Landsman is ambiguous with respect to the Applet Tags including a reference to the Advertising Management System.

(16) We now turn to our first question whether or not the applet tags include a reference to the Ad Management System of the advertising tag. In the specification Landsman makes the following statement at Col 20 lines 6 to 14:

As indicated by block 125, the browser then executes the interpreted code for the script which, in turn, dynamically writes applet tags --in the manner generally shown in FIGS. 2A and 2B and described above--into web page 35 in lieu of the advertising tag. These tags, which collectively form Transition Sensor applet 210, include a reference to a specific ad management system as specified in the second portion of advertising tag 40.

(17) From this passage it appears that the applet tags include a reference to the URL of an AMS. However, Table 2, which is an example of the rewritten applet tags, does not include a reference to a URL of an AMS. There are several components in Table 2, which reference the executable for the Transition Sensor applet, and the specific class and archive in which the applet is stored.

Declaration of Charles R. Martin

7

(18) The only portion, which I find may be indicating an AMS, is the following value in Table 2:

*<param name="adURL"
value="http://www.unicast.com/media/fireworks_01_ad_descriptor.txt">*

The 'adURL' value may be intended to indicate a specific AMS. However, there remains some confusion with how this value technically indicates the AMS. First, I must note that the URL in this value is the URL of "Unicast". Landsman uses Unicast as an example for an 'agent server' which hosts the AdController agent, including the JavaScript file, the Transition Sensor applet and the AdController applet. (Note the URL for the JavaScript file in Table 1 and the URL for the Transition Sensor applet in Table 2). In contrast, Landsman describes the AMS as a third-party server, which assumingly is different than the agent server (see e.g. Col 10 lines 3-22).

(19) Furthermore, the value assigned to 'adURL' indicates not only a URL, but also the filename of an AdDescriptor file. This is in contrast to what is clearly understood from Landsman's disclosure that the AMS selects the appropriate AdDescriptor file (See Col 10 lines 10-22, Col 20 line 55 to Col 21 line 26). More specifically, the AdController places a request for an AdDescriptor file from the AMS, and the AMS selects the AdDescriptor file. Thus, there is no need for either the Transition Sensor applet or the AdController applet to make reference to a particular AdDescriptor file. I therefore fail to understand the purpose of the applet tags' reference to an "AdDescriptor" file.

(20) However, I should note that as previously explained, the AdController applet at the client PC cannot directly request an AdDescriptor file from the AMS, rather it requests the ad from the agent server and the agent server serves as a proxy between the client PC and the AMS. As such, the above AdDescriptor may possibly

Declaration of Charles R. Martin

serve the function of identifying the AMS to the agent server.

(21) In any event, based on Table 2 as an example of the applet tags, I conclude that there's an inherent ambiguity with the above-quoted statement - that the applet tags include a reference to the AMS.

The JavaScript file does not inherently include a reference to the Advertising Management System.

(22) I now turn to the next question of whether or not the JavaScript file includes a reference to the Advertising Management System. The answer to this question is "no", as I hereby explain. As noted above the advertising tag of Table 1 includes two components. The first component is the reference to the JavaScript file (loadad.js). The second is a variable named "AdServer" which is assigned the URL of the AMS. 'Variables' are one of the most basic functions in most programming languages. Generally speaking, a variable is a handle by which static programming code may reference to dynamically changing attributes or values. In Landsman, the JavaScript file is a static file, which may be used across many web pages and websites. The JavaScript file may include script that refers to the variable AdServer, and may perform various functions without any need to internally identify the value AdServer. The value of AdServer is assigned externally by the web page linking to the JavaScript file. The value of AdServer may dynamically change depending on the web page or website that is referring to it. Different web pages or websites may assign different AMS's. When the browser loads the JavaScript file, it first 'initializes' (i.e. determines) and loads the value for the variable AdServer into the memory of the client PC. Thereafter, it parses the JavaScript file with the value of AdServer being attributed the value from the memory.

Declaration of Charles R. Martin

9

(23) As part of its script the JavaScript file may instruct the browser to output the current value of AdServer into the dynamically written applet tags. In such case, the applet tags would be identifying the reference to the AMS, although the JavaScript file itself would not include the identity of the AMS, rather it would obtain the AMS by determined the value assigned to the variable AdServer by the advertising tag.

(24) As I previously noted, there is some confusion with respect to whether or not the applet tags include a reference to the AMS. On one hand the specification clearly states that the applet tags do include the reference, while on the other hand, Table 2, which exemplifies the applet tags, does not include such a reference. However, even if we shall assume that the AMS is included in the applet tags, it does not necessarily follow that the JavaScript also includes the reference to the AMS. To the contrary, the fact that a reference to the AMS is included in the advertising tag and is assigned to the value AdServer, suggests that the JavaScript file does not identify the AMS, otherwise there would be no purpose in including this reference in the advertising tag.

The Transition Sensor Applet does not inherently include a reference the Advertising Management System.

(25) Lastly, we note based on the foregoing, that the Transition Sensor Applet also does not include a reference to the AMS. Although the passage that has been quoted in paragraph 16 of this declaration states that the TSA includes a reference to the AMS, we have already explained why that passage is misleading and erroneously refers to the applet tags as the TSA or as an 'applet'. As has been clarified the Transition Sensor applet and the applet tags are separate elements. Thus, that passage alone cannot provide any reasonable inference that the Transition Sensor applet includes a reference to the AMS.

(26) Moreover, Landsman clearly states that the Transition Sensor applet

obtains the URL of the AMS from the advertising tag, which suggests that the URL is not identified internally by the applet (See Col 34 lines 7-10, Fig 11 numeral 1130). Landsman also states that the TSA is persistent across many web pages and websites and over subsequent browser sessions, further suggesting that a single TSA can be used for different web pages, regardless of the AMS referred to by the advertising tag of each web page (Col 12 lines 50-62, Col 22 line 66 to Col 23 line 12). As such, I conclude that Landsman disclosure as a whole suggests that the URL of the AMS is not included in the TSA.

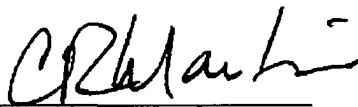
FINDINGS

(27) In view of the foregoing, I conclude the following:

1. The Transition Sensor applet and 'applet tags' are separate elements.
2. The Landsman Patent is ambiguous with respect to whether or not the applet tags include a reference to the AMS.
3. The Landsman Patent as a whole suggests that the JavaScript file does not include a reference to the AMS.
4. The Landsman Patent as a whole suggests that the Transition Sensor applet does not include a reference to the AMS.

I declare under penalty of perjury of the laws of the United States that the foregoing is true and correct.

Dated this 4 day of December, 2004, at Broomfield, Colorado.

Signature: 

Declaration of Charles R. Martin